

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of

Asuka MANKI

Serial No. (unknown)

Filed herewith

METHOD, DEVICE AND RECORD MEDIUM FOR SCOPE PROCESSING

**CLAIM FOR FOREIGN PRIORITY UNDER 35 U.S.C. 119
AND SUBMISSION OF PRIORITY DOCUMENT**

Assistant Commissioner for Patents

Washington, D.C. 20231

Sir:

Attached hereto is a certified copy of applicant's corresponding patent application filed in Japan under 2000-162470, on May 31, 2000.

Applicant herewith claims the benefit of the priority filing date of the above-identified application for the above-entitled U.S. application under the provisions of 35 U.S.C. 119.

Respectfully submitted,

YOUNG & THOMPSON

By

Benoît Castel

Benoît Castel
Attorney for Applicant
Customer No. 000466
Registration No. 35,041
745 South 23rd Street
Arlington, VA 22202
703/521-2297

May 31, 2001

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

Jc979 U.S. PTO
09/867421
05/31/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 5月31日

出 願 番 号

Application Number:

特願2000-162470

出 願 人

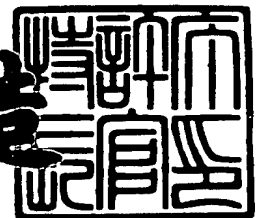
Applicant (s):

日本電気株式会社

2001年 3月23日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3023297

【書類名】 特許願

【整理番号】 45701595

【提出日】 平成12年 5月31日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/00
G06F 13/00

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 萬木 あす香

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100103090

【弁理士】

【氏名又は名称】 岩壁 冬樹

【電話番号】 03-3811-3561

【手数料の表示】

【予納台帳番号】 050496

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 スコープ処理方法およびOSI管理システムにおける管理対象装置

【特許請求の範囲】

【請求項1】 包含木上に格納された管理対象情報であるインスタンスを包含木から抽出するスコープ処理方法であって、スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動しながら目的とするインスタンスを抽出することを特徴とするスコープ処理方法。

【請求項2】 スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動するとき、開始位置から下位のインスタンスへの移動を試み、

下位のインスタンスに移動できなくなった場合には一つ横のインスタンスに移動してから下位のインスタンスへの移動を試みることを繰り返し、

一つ横のインスタンスに移動できなくなった場合には一つ上位のインスタンスに移動してから一つ横のインスタンスに移動してさらに下位のインスタンスへの移動を試みることを繰り返す

ことを特徴とする請求項1記載のスコープ処理方法。

【請求項3】 スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動するとき、下位のインスタンスに移動できなくなった場合と一つ横のインスタンスに移動できずに一つ上位のインスタンスに移動した場合に包含木内の移動を中断し、

移動を中断したインスタンスが目的とするインスタンスであるか否かを判断し

移動を中断したインスタンスが目的とするインスタンスであるならばそのインスタンスを抽出して移動を再開する

ことを特徴とする請求項2記載のスコープ処理方法。

【請求項4】 スコープ処理の開始位置からの相対深度に対応する記憶領域を有するスタック領域を保持し、

一つ下位のインスタンスに移動したときには移動後のインスタンスの相対深度

に対応する記憶領域に移動後のインスタンスの情報を記憶し、

一つ横のインスタンスに移動したときには現在位置の相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、

一つ上位のインスタンスに移動したときには移動前のインスタンスの相対深度に対応する記憶領域に記憶している情報を削除し、

インスタンスを抽出するときには現在位置の相対深度に対応する記憶領域からインスタンスの情報を抽出する

ことを特徴とする請求項 3 記載のスコープ処理方法。

【請求項 5】 包含木上に格納された管理対象情報は、OSI 管理システムにおける管理対象装置の情報であることを特徴とする請求項 1 ないし請求項 4 記載のスコープ処理方法。

【請求項 6】 管理対象情報を示すインスタンスを包含木上に格納する管理情報ベースを備えた OSI 管理システムにおける管理対象装置であって、スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動しながら目的とするインスタンスを抽出するインスタンス抽出手段を備えたことを特徴とする OSI 管理システムにおける管理対象装置。

【請求項 7】 インスタンス抽出手段は、包含木内で開始位置から下位のインスタンスへの移動を試み、

下位のインスタンスに移動できなくなった場合には一つ横のインスタンスに移動してから下位のインスタンスへの移動を試みることを繰り返し、

一つ横のインスタンスに移動できなくなった場合には一つ上位のインスタンスに移動してから一つ横のインスタンスに移動してさらに下位のインスタンスへの移動を試みることを繰り返す

ことを特徴とする請求項 6 記載の OSI 管理システムにおける管理対象装置。

【請求項 8】 インスタンス抽出手段は、包含木内で下位のインスタンスに移動できなくなった場合と一つ横のインスタンスに移動できずに一つ上位のインスタンスに移動した場合に包含木内の移動を中断し、

移動を中断したインスタンスが目的とするインスタンスであるか否かを判断し

移動を中断したインスタンスが目的とするインスタンスであるならばそのインスタンスを抽出して移動を再開する

ことを特徴とする請求項 7 記載の O S I 管理システムにおける管理対象装置。

【請求項 9】 インスタンス抽出手段は、スコープ処理の開始位置からの相対深度に対応する記憶領域を有するスタック領域を保持し、

一つ下位のインスタンスに移動したときには移動後のインスタンスの相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、

一つ横のインスタンスに移動したときには現在位置の相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、

一つ上位のインスタンスに移動したときには移動前のインスタンスの相対深度に対応する記憶領域に記憶している情報を削除し、

インスタンスを抽出するときには現在位置の相対深度に対応する記憶領域からインスタンスの情報を抽出する

ことを特徴とする請求項 8 記載の O S I 管理システムにおける管理対象装置。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、インスタンスを抽出するスコープ処理方法および、短時間でスコープ処理を行い、管理情報ベースに保持すべき情報量を少なくする O S I 管理システムにおける管理対象装置に関する。

【 0 0 0 2 】

【従来の技術】

図 1 1 は、O S I (Open Systems Interconnection) 管理システムの構成の例を示すブロック図である。O S I 管理システムにおいて、管理対象装置 1 0 0 は、管理対象であるネットワーク上のネットワークノードである。具体的には、管理対象ネットワーク上に存在するパーソナルコンピュータ、プリンタ等の資源が管理対象装置 1 0 0 に該当する。ネットワーク管理システム 1 1 0 は、管理対象装置 1 0 0 を監視し制御する。管理対象装置 1 0 0、ネットワーク管理システム 1 1 0 の制御部 1 0 1、1 1 1 は、いずれも記憶装置 1 0 2、1 1 2 に記憶するブ

プログラムに従い、各装置を制御する。また、インタフェース部 1 0 4, 1 1 4 は、ネットワークを介して情報の送受信を行う。

【 0 0 0 3 】

M I B (Management Information Base: 管理情報ベース) 7 は、管理対象装置 1 0 0 を表した管理オブジェクトを木構造として格納している。この木構造をなす木を包含木という。各管理オブジェクトは、管理対象装置 1 0 0 の様々な情報を示すインスタンスを有する。

【 0 0 0 4 】

エージェント 1 0 3 およびマネージャ 1 1 3 は、それぞれ管理対象装置 1 0 0、ネットワーク管理システム 1 1 0 におけるネットワーク管理処理を行うソフトウェアモジュールである。実際に動作するのは制御部 1 0 1, 1 1 1 であるが、以下の説明ではソフトウェアの動作として説明する。O S I 管理システムでは、エージェント 1 0 3 およびマネージャ 1 1 3 が C M I P (Common Management Information Protocol) により情報の授受を行うことで、ネットワークを管理している。

【 0 0 0 5 】

従来、エージェント 1 0 3 が、M I B 7 内に存在するインスタンスを絞り込んで抽出する処理 (スコープ処理) を行う場合、次のような方法を用いていた。まず、包含木のノードに格納されている各インスタンスは、あらかじめ、包含木の根 (ルート) からの深度情報とともに格納されていた。スコープ処理の際、エージェント 1 0 3 は、抽出すべきインスタンスの条件 (深度に関する条件) と各インスタンスが保持している深度情報とを比較し、条件に合致する深度情報を保持するインスタンスを抽出していた。

【 0 0 0 6 】

【発明が解決しようとする課題】

しかし、このような方法では、エージェント 1 0 3 は、条件に合致する深度情報を保持しているか否かの検査を包含木内の全インスタンスについて行わねばならず、目的とするインスタンスの抽出完了までに時間がかかっていた。また、包含木内の各インスタンスは、それぞれ深度情報とともに格納されていたため、包

含木情報の占める領域が大きくなっていた。

【0007】

本発明は、短時間でインスタンスを抽出するスコープ処理方法と、短時間でスコープ処理を行い、MIBに保持すべき情報量を少なくするOSI管理システムにおける管理対象装置を提供することを目的とする。

【0008】

【課題を解決するための手段】

本発明によるスコープ処理方法は、包含木上に格納された管理対象情報であるインスタンスを包含木から抽出するスコープ処理方法であって、スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動しながら目的とするインスタンスを抽出することを特徴とする。

【0009】

スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動するときに、例えば、開始位置から下位のインスタンスへの移動を試み、下位のインスタンスに移動できなくなった場合には一つ横のインスタンスに移動してから下位のインスタンスへの移動を試みることを繰り返し、一つ横のインスタンスに移動できなくなった場合には一つ上位のインスタンスに移動してから一つ横のインスタンスに移動してさらに下位のインスタンスへの移動を試みることを繰り返す。このようなスコープ処理方法によれば、目標とするインスタンスを短時間で抽出することができる。

【0010】

スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動するときに、例えば、下位のインスタンスに移動できなくなった場合と一つ横のインスタンスに移動できずに一つ上位のインスタンスに移動した場合に包含木内の移動を中断し、移動を中断したインスタンスが目的とするインスタンスであるか否かを判断し、移動を中断したインスタンスが目的とするインスタンスであるならばそのインスタンスを抽出して移動を再開する。

【0011】

また、スコープ処理の開始位置からの相対深度に対応する記憶領域を有するス

タック領域を保持し、一つ下位のインスタンスに移動したときには移動後のインスタンスの相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、一つ横のインスタンスに移動したときには現在位置の相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、一つ上位のインスタンスに移動したときには移動前のインスタンスの相対深度に対応する記憶領域に記憶している情報を削除し、インスタンスを抽出するときには現在位置の相対深度に対応する記憶領域からインスタンスの情報を抽出する。このようなスコープ処理方法によれば、スタック領域に記憶する情報に基づいてどの深度のインスタンスを参照しているのかを判断できるので、包含木が深度情報を保持する必要がない。

【 0 0 1 2 】

また、包含木上に格納された管理対象情報は、例えば、OSI 管理システムにおける管理対象装置の情報である。

【 0 0 1 3 】

また、本発明による OSI 管理システムにおける管理対象装置は、管理対象情報を示すインスタンスを包含木上に格納する管理情報ベースを備えた OSI 管理システムにおける管理対象装置であって、スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動しながら目的とするインスタンスを抽出するインスタンス抽出手段を備えたことを特徴とする。

【 0 0 1 4 】

インスタンス抽出手段は、例えば、包含木内で開始位置から下位のインスタンスへの移動を試み、下位のインスタンスに移動できなくなった場合には一つ横のインスタンスに移動してから下位のインスタンスへの移動を試みることを繰り返す、一つ横のインスタンスに移動できなくなった場合には一つ上位のインスタンスに移動してから一つ横のインスタンスに移動してさらに下位のインスタンスへの移動を試みることを繰り返す。このような管理対象装置によれば、目標とするインスタンスを短時間で抽出することができる。

【 0 0 1 5 】

また、インスタンス抽出手段は、例えば、包含木内で下位のインスタンスに移動できなくなった場合と一つ横のインスタンスに移動できずに一つ上位のインス

タンスに移動した場合に包含木内の移動を中断し、移動を中断したインスタンスが目的とするインスタンスであるか否かを判断し、移動を中断したインスタンスが目的とするインスタンスであるならばそのインスタンスを抽出して移動を再開する。

【0016】

また、インスタンス抽出手段は、スコープ処理の開始位置からの相対深度に対応する記憶領域を有するスタック領域を保持し、一つ下位のインスタンスに移動したときには移動後のインスタンスの相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、一つ横のインスタンスに移動したときには現在位置の相対深度に対応する記憶領域に移動後のインスタンスの情報を記憶し、一つ上位のインスタンスに移動したときには移動前のインスタンスの相対深度に対応する記憶領域に記憶している情報を削除し、インスタンスを抽出するときには現在位置の相対深度に対応する記憶領域からインスタンスの情報を抽出する。このような管理対象装置によれば、スタック領域に記憶する情報に基づいてどの深度のインスタンスを参照しているのかを判断できるので、包含木が深度情報を保持する必要がない。

【0017】

【発明の実施の形態】

以下、本発明の実施の形態を図面を参照して説明する。

図11に示すようなOSI管理システムにおいて、MIB7が格納する包含木からインスタンスを抽出するスコープ処理を説明する。包含木は、ルートを持ち、ルートから木構造として格納される各オブジェクトへのポインタが張られている。図1は、スコープ処理を行うスコープ記述子オブジェクトを示す説明図である。スコープ記述子オブジェクト2は、オブジェクト指向技術におけるオブジェクトであり、包含木からインスタンスを抽出する。

【0018】

スコープ記述子オブジェクト2において、スコープ条件情報部3はスコープ条件等を保持する。図2は、スコープ条件情報部3を示す説明図である。開始位置部10は、開始位置を記憶する。開始位置は、スコープ処理の際に包含木内のど

のインスタンスにスコープ条件をあてはめるかを示す情報である。例えば、開始位置部 1 0 が図 3 に示す包含木における B - 1 を開始位置として保持するならば、B - 1 にスコープ条件をあてはめたインスタンスを抽出することになる。

【 0 0 1 9 】

スコープ条件部 1 1 は、以下に示す各スコープ条件の何れかを記憶する。

” baseObject ” は、開始位置にあるインスタンスを指す条件である。” firstLevelOnly ” は、開始位置より一つ下位の深度のインスタンスを指す条件である。

” wholeSubtree ” は、開始位置以下の全インスタンスを指す条件である。” individualLevels ” は、開始位置から指定した深度にあるインスタンスを指す条件である。また、” baseToNthLevel ” は、開始位置から指定した深度までにある全インスタンスを指す条件である。例えば、開始位置が B - 1 であり、スコープ条件が ” firstLevelOnly ” であるならば、スコープ記述子オブジェクト 2 は、B - 1 よりも一つ下位の深度のインスタンス、すなわち D - 1 ないし D - 4 を抽出する。

【 0 0 2 0 】

開始深度部 1 2 は、開始深度を記憶する。開始深度は、スコープ条件が定める範囲内で一番浅い深度を開始位置からの相対深度として示す情報である。終了深度部 1 3 は、終了深度を記憶する。終了深度は、スコープ条件が定める範囲内で一番深い深度を開始位置からの相対深度として示す情報である。例えば、スコープ条件 ” firstLevelOnly ” は、開始位置より一つ下位の深度を指定するので、開始深度および終了深度はともに 1 となる。

【 0 0 2 1 】

また、スコープ処理振舞部 4 は、開始深度および終了深度が示す範囲内のインスタンスを抽出する関数へのポインタを保持する。この関数は、参照するインスタンスを変えていき、スコープ条件によって定められるインスタンスを抽出する。参照するインスタンスを変えることを「移動する」と記す。スコープ記述子オブジェクト 2 は、この関数を呼び出して包含木内を移動し、インスタンスを抽出する。また、スコープ処理振舞部 4 は、包含木内を移動する際の移動方法（「下に行く」、「横に行く」等）を保持する。移動方法は、スコープ記述子オブジェ

クト生成時に指定されるが、スコープ処理の過程において変更される。また、スコープ処理振舞部 4 は、包含木内を移動するときに、現在の深度に関する情報等を保持する。

【 0 0 2 2 】

スタック領域 5 は、包含木内の深度と 1 対 1 に対応する記憶領域を有する。深度に対応する各記憶領域は、包含木内の位置を示す現在位置情報とその位置におけるインスタンスの情報とを記憶する。スコープ記述子オブジェクト 2 が包含木内を横方向に移動するときは、その深度に対応する記憶領域において、現在位置情報とその位置におけるインスタンス情報を上書きしていく。また、包含木内を下方向に移動するときは、移動した位置における現在位置情報とインスタンス情報を移動先深度に対応する領域に書き込み、上方向に移動するときは、移動する前の深度に対応する領域の情報を削除する。

【 0 0 2 3 】

スコープ記述子オブジェクト 2 が移動するときのスタック領域 5 の変化の例を図 4 に示す。図 4 は、スコープ記述子オブジェクト 2 が深度 1 のインスタンスから下方向または上方向へ移動した場合におけるスタック領域 5 の変化を示している。深度 1 から深度 2 へ移動するときには、移動先深度である深度 2 の領域に現在位置情報とその位置におけるインスタンス情報を書き込む。また、深度 1 から深度 0 に移動するときには、深度 1 の領域に記憶していた情報を削除する。また、深度 1 において横方向に移動するときは、深度 1 の領域に記憶する情報を更新する。

【 0 0 2 4 】

次に、動作について説明する。

図 5 は、エージェント 1 0 3 がマネージャ 1 1 3 から所定のインスタンスを抽出する旨の命令を受けたときに、スコープ記述子オブジェクト 2 を生成する際の動作の例を示すフローチャートである。エージェント 1 0 3 は、マネージャ 1 1 3 から開始位置およびスコープ条件を入力し（ステップ S 5 1）、スコープ条件から開始深度および終了深度を計算する（ステップ S 5 2）。続くステップ S 5 3 では、これらの条件をスコープ条件情報部 3 に保持し、スコープ処理振舞部 4

およびスタック領域 5 を確保するスコープ記述子オブジェクト 2 (マネージャ 1 1 3 の命令に対応したスコープ記述子インスタンス) を生成する。ステップ S 5 3 において、スコープ処理振舞部 4 には「下に行く」という移動方法を保持させる。ステップ S 5 1 ~ S 5 3 の動作により、包含木内のインスタンスを抽出するスコープ記述子オブジェクト 2 が生成される。

【 0 0 2 5 】

図 6 は、スコープ記述子オブジェクト 2 が包含木内を移動して、開始位置およびスコープ条件によって定められるインスタンスを抽出する動作の例を示すフローチャートである。ステップ S 6 1 において、スコープ記述子オブジェクト 2 は、スコープ条件部 1 1 に保持するスコープ条件が "baseObject" であるか判断する。スコープ条件が "baseObject" であるならば、包含木内の開始位置のインスタンスを抽出し (ステップ S 6 2)、スコープ処理を終了する。

【 0 0 2 6 】

ステップ S 6 1 で、スコープ条件が "baseObject" でなかったならば、スコープ記述子オブジェクト 2 は、開始位置として指定されたインスタンスまで包含木内を移動する (ステップ S 6 3)。スコープ処理振舞部 4 が保持する移動方法は変更されていないので、ステップ S 6 3 において移動方法は「下に行く」として保持している。続いて、開始位置となるインスタンスから深さ優先探索方法に従い、包含木内を移動する。このとき、下位にインスタンスが無く横に移動するときと、横へ移動ができず上に移動したときに移動を中断する (ステップ S 6 4)。ステップ S 6 4 において包含木内を移動したときには、移動した方向に応じてスタック領域 5 に記憶する情報を更新する。ステップ S 6 4 における深さ優先探索方法による移動については後述する。

【 0 0 2 7 】

スコープ記述子オブジェクト 2 は、ステップ S 6 4 で移動を中断した後に、その位置が処理終了位置か否かを判断する (ステップ S 6 5)。処理終了位置か否かに関する情報 (探索終了フラグ) は、後述の図 7 に示すステップ S 7 9 で設定される。移動を中断した位置が処理終了位置であるならば、スコープ処理を終了する。

【 0 0 2 8 】

移動を中断した位置が処理終了位置でないならば、現在位置の深度と開始深度とを確認して比較する（ステップ S 6 6）。このとき、現在位置の深度は、開始位置からの相対深度として求める。現在位置の深度が開始深度より小さいならば、その位置からステップ S 6 4 の移動を再開する。また、現在位置の深度が開始深度以上であるならば、その位置におけるインスタンスを抽出し（ステップ S 6 7）、ステップ S 6 4 の移動を再開する。ステップ S 6 7 では、スタック領域 5 から現在位置のインスタンスを抽出する。

【 0 0 2 9 】

図 7 は、ステップ S 6 4 における深さ優先探索方法による移動の例を示したフローチャートである。まず、スコープ記述子オブジェクト 2 は、包含木内における次の移動方法を確認する（ステップ S 7 1）。移動方法が「下に行く」であるならば、包含木において一つ下位への移動を試みる（ステップ S 7 2）。ステップ S 7 2 において下に移動できた場合には、ステップ S 7 1 以降の動作を繰り返す。ステップ S 7 2 において下に移動できなかった場合には、現在位置が開始位置に戻っているかどうかを判断する（ステップ S 7 3）。開始位置に戻っていないければ、スコープ記述子オブジェクト 2 は、スコープ処理振舞部 4 が保持する移動方法を「横に行く」に変更して（ステップ S 7 4）、ステップ S 6 4 における移動を中断し、ステップ S 6 5 に進む。ステップ S 7 3 において、開始位置に戻っているならば、現在位置を処理終了位置として探索終了フラグを立て（ステップ S 7 9）、ステップ S 6 5 に進む。

【 0 0 3 0 】

ステップ S 7 1 において移動方法が「横に行く」であるならば、スコープ記述子オブジェクト 2 は、一つ横への移動を試みる（ステップ S 7 5）。ステップ S 7 5 において、開始位置から横への移動はできない。横に包含木が続いていて横に移動できたならば、次の移動方法を「下に行く」として（ステップ S 7 6）、ステップ S 7 1 以降の動作を繰り返す。ステップ S 7 5 において横に移動できなかった場合には、一つ上位への移動を試みる（ステップ S 7 7）。ステップ S 7 7 において上に移動できた場合には、次の移動方法を「横に行く」にして（ステ

ップ S 7 8)、移動を中断し、ステップ S 6 5に進む。ステップ S 7 7で上に移動できなかったならば、現在位置を処理終了位置として探索終了フラグを立て(ステップ S 7 9)、ステップ S 6 5に進む。

【 0 0 3 1 】

図 8 (a) は、ステップ S 7 2で下への移動を試みる場合における移動可否判断の例を示すフローチャートである。下への移動可否判断を行う場合、スコープ記述子オブジェクト 2 は、現在位置の深度および終了深度を確認する(ステップ S 8 1)。ただし、現在位置の深度は、開始位置からの相対深度として求める。続いて、スコープ記述子オブジェクト 2 は、現在位置の深度と終了深度とを比較し(ステップ S 8 2)、終了深度の方が大きいならば、ステップ S 7 2において下に移動する。終了深度が現在位置の深度以下であるならば、ステップ S 7 2において下に移動できず、ステップ S 7 3以降の動作を行う。ただし、ステップ S 8 2で終了深度の方が大きいと判断しても、現在位置のインスタンスに下位のインスタンスが存在しない場合がある。この場合には、ステップ S 7 2において下に移動できず、ステップ S 7 3以降の動作を行う。

【 0 0 3 2 】

図 8 (b) は、ステップ S 7 7で上への移動を試みる場合における移動可否判断の例を示すフローチャートである。上への移動可否判断を行う場合、スコープ記述子オブジェクト 2 は、現在位置が開始位置に戻っているかどうかを判断する(ステップ S 8 6)。現在位置に戻っていないならば、ステップ S 7 7で上に移動する。現在位置に戻っているならば、ステップ S 7 7において上に移動できず、ステップ S 7 9以降の動作を行う。

【 0 0 3 3 】

次に、各スコープ条件のもとでの動作を、具体例を用いて説明する。

M I B 7 は、図 3 に示す包含木を登録されているものとする。また、開始位置は B - 1 とする。

【 0 0 3 4 】

スコープ条件が、" baseObject" である場合、ステップ S 5 1 ~ S 5 3 の処理により生成されたスコープ記述子オブジェクト 2 は、ステップ S 6 1, S 6 2 の

処理を行う。すなわち、ステップ S 6 2 で開始位置 (B - 1) のインスタンスを抽出してスコープ処理を終了する。

【 0 0 3 5 】

次に、スコープ条件が "firstLevelOnly" である場合について説明する。

スコープ条件が定める範囲は B - 1 より一つ下位の深度である。したがって、エージェント 1 0 3 は、ステップ S 5 2 で開始深度および終了深度をそれぞれ 1 として計算し、ステップ S 5 3 でこれらの情報を保持するスコープ記述子オブジェクト 2 を生成する。

【 0 0 3 6 】

スコープ記述子オブジェクト 2 は、ステップ S 6 1 の後、開始位置である B - 1 に移動する (ステップ S 6 3)。続いて、深さ優先探索方法で包含木内を移動する (ステップ S 6 4)。ここで、最初の移動方法を「下に行く」として保持しているので、ステップ S 7 1 の後、下への移動を試みる (ステップ S 7 2)。終了深度は 1、現在位置 (B - 1) の相対深度は 0 であるので、スコープ記述子オブジェクト 2 は D - 1 に移動できる。スコープ記述子オブジェクト 2 は、D - 1 に移動したときに、スタック領域 5 の深度 1 に対応する領域に現在位置およびその位置におけるインスタンスの情報を記憶する。続いて、再度ステップ S 7 1、S 7 2 の動作を行う。現在位置が D - 1 となった場合、終了深度および現在位置の深度はともに 1 であるので、スコープ記述子オブジェクト 2 は、これより下に移動できない。したがって、ステップ S 7 3、S 7 4 の動作を行い、移動を中断する。

【 0 0 3 7 】

続いて、ステップ S 6 5、S 6 6 の判断を行う。現在位置 (D - 1) の相対深度は 1、開始深度は 1 であるので、ステップ S 6 6 の後にステップ S 6 7 の処理を行う。スコープ記述子オブジェクト 2 は、スタック領域 5 の深度 1 に対応する領域から D - 1 のインスタンスを抽出し (ステップ S 6 7)、再度 D - 1 から移動を開始する (ステップ S 6 4)。

【 0 0 3 8 】

ステップ S 7 4 で移動方向を「横に行く」に変更したため、ステップ S 7 1 の

後、横への移動を試みる（ステップ S 7 5）。このとき、D-1 から D-2 に移動したスコープ記述子オブジェクト 2 は、スタック領域 5 の深度 1 に対応する領域に D-2 の情報を上書きする。スコープ記述子オブジェクト 2 は、移動方向を「下に行く」に変更して（ステップ S 7 6）、この後、ステップ S 7 1, S 7 2, S 7 3, S 7 4 の動作を行う。（ステップ S 7 2 において、D-2 の相対深度と終了深度はともに 1 であるので、ステップ S 7 2 の後にはステップ S 7 3 の動作を行うことになる。）

【 0 0 3 9 】

この後、移動を中断し、ステップ S 6 5, S 6 6 の判断をして、ステップ S 6 7 で D-2 のインスタンスを抽出する。D-3, D-4 の抽出も、同様の動作で行う。D-4 を抽出後に移動を再開したとき、ステップ S 7 5 では横に行けないため、上への移動（B-1）への移動を試みる（ステップ S 7 7）。現在位置が D-4 である場合、開始インスタンスに戻っていないので、上（B-1）に移動し、ステップ S 7 8 の後に移動を中断する。続いて、ステップ S 6 5, S 6 6 の判断を行うが、B-1 の相対深度は 0、開始深度は 1 なので、ステップ S 6 6 の後にはステップ S 6 7 を行わずに移動を再開する。すなわち、B-1 のインスタンスは抽出されない。

【 0 0 4 0 】

移動再開後、開始位置から横への移動はできないので、ステップ S 7 1, S 7 5, S 7 7 の動作を行う。ステップ S 7 7 において、現在位置 B-1 に戻っているので上に移動できず、現在位置を処理終了位置として（ステップ S 7 9）、移動を中断する。その後、ステップ S 6 5 の判断により、スコープ処理を終了する。この結果、D-1、D-2、D-3、D-4 のインスタンスが抽出される。

【 0 0 4 1 】

次に、スコープ条件が” wholeSubtree” である場合について説明する。

この場合、スコープ条件が定める範囲は B-1 以下の全てのインスタンスであるので、エージェント 1 0 3 は、開始深度を 0、終了深度を 2 として保持するスコープ記述子オブジェクト 2 を生成する（ステップ S 5 1 ~ S 5 3）。

【 0 0 4 2 】

スコープ記述子オブジェクト 2 は、" firstLevelOnly" の場合と同様にステップ S 6 3 において B - 1 に移動した後、さらに D - 1 に移動し、スタック領域 5 を更新する。D - 1 に移動したときの深度は 1 であり、終了深度は 2 であるので、D - 1 の下位にインスタンスが存在すればさらに下位に移動できる。しかし、D - 1 の下にはインスタンスが存在しないので、D - 1 より下に移動できず、ステップ S 7 3, S 7 4 を行い移動を中断する。その後、" firstLevelOnly" の場合と同様にステップ S 6 5, S 6 6 の判断を行い、ステップ S 6 7 において D - 1 のインスタンスをスタック領域 5 から抽出する。

【 0 0 4 3 】

スコープ記述子オブジェクト 2 は、ステップ S 7 4 で移動方向を「横に行く」に変更したため、移動を再開したときにステップ S 7 1, S 7 5 により D - 2 に移動する。その後移動方向を「下に行く」に変更して（ステップ S 7 6）、この後、ステップ S 7 1, S 7 2 の動作を繰り返す。D - 2 の相対深度は 1、終了深度は 2 であるのでステップ S 7 2 において G - 1 に移動する。このとき、スタック領域 5 の深度 2 に対応する領域に現在位置およびその位置におけるインスタンスの情報を記憶する。さらに、ステップ S 7 1, S 7 2 を繰り返すときには、G - 1 の相対深度は 2 であるので下位に移動できず、ステップ S 7 3, S 7 4 の後、移動を中断する。そして、ステップ S 6 5, S 6 6 の判断の後、G - 1 のインスタンスを抽出する（ステップ S 6 7）。

【 0 0 4 4 】

G - 1 のインスタンスを抽出後、移動を再開した場合、ステップ S 7 1, S 7 2 により横への移動を試みる。しかし、G - 1 の横にはインスタンスがなく移動できないため、ステップ S 7 7 で上への移動を試みる。現在位置が G - 1 である場合、開始位置に戻っていないので、G - 1 から上のインスタンス（D - 2）に移動する。このとき、スタック領域 5 の深度 2 に対応する領域の情報を削除する。続いて、ステップ S 7 8 において、移動方法を「横に行く」に変更し、移動を中断する。そして、ステップ S 6 5, S 6 6 の判断の後、D - 2 のインスタンスを抽出する（ステップ S 6 7）。

【 0 0 4 5 】

G-2, D-3, D-4の抽出も、同様の動作で行う。D-4を抽出後に移動を再開したとき、ステップS75では横に行けないため、上への移動(B-1)への移動を試みる(ステップS77)。現在位置がD-4である場合、開始位置に戻っていないので、上(B-1)に移動し、ステップS78の後に移動を中断する。このとき、スタック領域5の深度1に対応する領域の情報を削除する。そして、ステップS65, S66の判断の後、B-1のインスタンスを抽出する(ステップS67)。

【0046】

更に移動を再開したときには、ステップS71, S75, S77の処理を行うが、ステップS77において、開始位置(B-1)に戻っているため、上に移動できず、ステップS79で現在位置を終了位置とし、ステップS65の判断によってスコープ処理を終了する。この結果、D-1ないしD-4、およびB-1の各インスタンスが全て抽出される。

【0047】

次に、スコープ条件が"individualLevels"で、深度2を指定した場合について説明する。この場合、スコープ条件が定める範囲はB-1から二つ下位の深度であるため、エージェント103は、開始深度を2、終了深度を2として保持するスコープ記述子オブジェクト2を生成する(ステップS51~S53)。

【0048】

スコープ記述子オブジェクト2は、ステップS63でB-1に移動した後、B-1から下へ移動する。そして、"wholeSubtree"の場合と同様にD-1に移動し、ステップS73, S74にのち、移動を中断する。その後、ステップS65, S66の判断を行うが、ステップS66において、D-1の相対深度は1、開始深度は2であるため、ステップS67の処理を行わずに移動を再開する。

【0049】

移動を再開したときには、"wholeSubtree"の場合と同様に、G-1まで移動して移動を中断する。その後のステップS65, S66の判断を行うが、ステップS66において、G-1の相対深度は2、開始深度は2であるため、ステップS67の処理を行い、G-1のインスタンスを抽出する。続いて移動を再開した

ときにも、“wholeSubtree”の場合と同様にD-2で移動を中断する。このとき、ステップS 6 6において、D-2の相対深度は1、開始深度は2であるので、ステップS 6 7の処理を行わずに移動を再開する。

【0 0 5 0】

移動再開後も同様に処理を進める。G-2、D-3、D-4において移動を中断するが、ステップS 6 6の判断により、D-3とD-4についてはインスタンスが抽出されない。また、D-4の位置から移動を再開した後は、“firstLevelOnly”の場合と同様にD-4からB-1に移動し、B-1で移動を中断する。そして、B-1で移動を中断したときには、ステップS 6 6の判断によりB-1のインスタンスを抽出しない。その後、B-1から移動を再開した後の動作も、“firstLevelOnly”の場合と同様である。この結果、G-1とG-2が抽出される。

【0 0 5 1】

次に、スコープ条件が“baseToNthLevel”で、深度1を指定した場合について説明する。この場合、スコープ条件が定める範囲はB-1からB-1の一つ下位の深度までであるので、エージェント1 0 1は、開始深度を0、終了深度を1として保持するスコープ記述子オブジェクト2を生成する（ステップS 5 1～S 5 3）。

【0 0 5 2】

スコープ記述子オブジェクト2は、ステップS 6 3でB-1に移動した後、B-1から下へ移動する。このとき、終了深度は1であるので深度2以下のインスタンスへは移動しない。D-1に移動し、ステップS 7 3、S 7 4の後、移動を中断する。その後、ステップS 6 5、S 6 6の判断を行う。ステップS 6 6においてD-1の相対深度は1、開始深度は0であるので、ステップS 6 7でD-1を抽出する。移動再開後は、他のスコープ条件の場合と同様にD-2に移動する。しかし、終了深度が1であるので、D-2の下位のG-1には移動せず、ステップS 7 3、S 7 4の後、移動を中断し、D-2を抽出する。同様にD-3、D-4を抽出し、D-4抽出後は“wholeSubtree”の場合と同様にB-1を抽出してスコープ処理を終了する。

【 0 0 5 3 】

ここでは、図 3 に示す包含木中の B - 1 を開始位置とした場合を例に説明したが、他の場合に関しても同様に目的のインスタンスを抽出できる。また、スコープ条件として” individualLevels” や” baseToNthLevel” とともに指定する深度を変えた場合にも、その深度に応じたインスタンスを抽出できる。

【 0 0 5 4 】

以上のようなスコープ処理方法によれば、スコープ記述子オブジェクト 2 は、開始位置とスコープ条件で定まる範囲内でインスタンスを移動し、処理対象外のインスタンスへは移動しない。したがって、包含木上の全インスタンスを移動していく必要がないので、スコープ処理時間が少なくて済む。また、スコープ記述子オブジェクト 2 は、どの深度のインスタンスを参照しているのかを判断できるので、M I B 7 は、包含木情報として各インスタンスの深度情報を保持する必要がない。

【 0 0 5 5 】

次に、本発明の他の実施の形態について説明する。この実施の形態によるスコープ処理方法は、開始位置およびスコープ条件によって定められるインスタンスを抽出する際、特定の属性を有するインスタンスを抽出するスコープ処理方法である。図 9 は、この実施の形態においてスコープ処理を行うスコープ記述子オブジェクトを示す。このスコープ記述子オブジェクト 2 0 は、スコープ条件情報部 3、スコープ処理振舞部 4、スタック領域 5 の他に、フィルタ情報部 2 1 を有し、フィルタ情報部 2 1 に特定のインスタンスのみを抽出するためのフィルタ条件を保持する。

【 0 0 5 6 】

このスコープ記述子オブジェクト 2 0 を生成する際の動作は、図 5 に示す動作と同様である。ただし、エージェント 1 0 3 は、ステップ S 5 1 においてスコープ条件と開始位置だけでなくフィルタ条件も入力し、このフィルタ条件をフィルタ情報部 2 1 に保持するスコープ記述子オブジェクト 2 0 を生成する。

【 0 0 5 7 】

図 1 0 は、スコープ記述子オブジェクト 2 0 が、包含木内を移動して開始位置

、スコープ条件およびフィルタ条件によって定められるインスタンスを抽出する動作の例を示すフローチャートである。このフローチャートにおいて、ステップ S 6 1～S 6 7 の動作は、図 6 に示す場合と同様である。ただし、ステップ S 6 6 の後に、スコープ記述子オブジェクト 20 は、現在位置（移動を中断した位置）のインスタンスの属性がフィルタ情報部 21 に保持するフィルタ条件に合致しているかを検査するフィルタ処理を行う（ステップ S 6 8）。したがって、この後に行うステップ S 6 7 では、フィルタ条件に合致するインスタンスのみが抽出されることになる。また、ステップ S 6 1 において、スコープ条件が”baseObject”であると判断したならば、スコープ記述子オブジェクト 20 は、開始位置のインスタンスの属性がフィルタ情報部 21 に保持するフィルタ条件に合致しているかを検査するフィルタ処理を行う（ステップ S 6 9）。したがって、この後に行うステップ S 6 2 では、開始位置のインスタンスがフィルタ条件に合致しないならば抽出されないことになる。

【0058】

この実施例における具体的な動作を説明する。

MI B 7 には図 3 に示す包含木が登録されており、D-1, D-2 のみが”属性 A”を有しているものとする。また、ステップ S 5 1 では、開始位置として B-1 を入力し、フィルタ条件として”属性 A”を入力するものとする。

【0059】

スコープ条件が”firstLevelOnly”である場合について説明する。

この場合、スコープ記述子オブジェクト 20 は、ステップ S 6 4 において前述の実施例と同様に深さ優先探索方法による移動を行う。移動を中断すると、ステップ S 6 5, S 6 6, S 6 8, S 6 7 の動作を行い、ステップ S 6 4 の移動を再開する。図 7 に示す深さ優先探索方法による移動は、前述の実施例と同様であるので、この場合、スコープ記述子オブジェクト 20 は、D-1, D-2, D-3, D-4 において移動を中断する。中断する度に、ステップ S 6 8 のフィルタ処理が行われるが、D-1, D-2 は”属性 A”を有しているので、D-1 や D-2 で移動が中断された場合、これらのインスタンスはフィルタ処理を通過する。したがって、スコープ記述子オブジェクト 20 は、ステップ S 6 7 において D-

1 や D - 2 のインスタンスを抽出する。

【 0 0 6 0 】

しかし、” 属性 A ” を有しない D - 3 や D - 4 で移動を中断した場合、D - 3 , D - 4 のインスタンスは、ステップ S 6 8 のフィルタ処理を通過しない。よって、スコープ記述子オブジェクト 2 0 は、ステップ S 6 7 において D - 3 や D - 4 のインスタンスを抽出しない。

【 0 0 6 1 】

スコープ条件が、” wholeSubtree ”、” individualLevels ”、” baseToNthLevel ” の場合も、ステップ S 6 7 でインスタンスを抽出する前に、フィルタ処理（ステップ S 6 8）を行うので、フィルタ条件に合致するインスタンスのみを抽出することになる。

【 0 0 6 2 】

スコープ条件が ” baseObject ” である場合、ステップ S 6 1 の判断の後、開始位置のインスタンスにフィルタ処理を行い（ステップ S 6 9）、開始位置のインスタンスを抽出する（ステップ S 6 2）。しかし、B - 1 は ” 属性 A ” を有しておらず、ステップ S 6 9 のフィルタ処理を通過しないので、ステップ S 6 2 では抽出されない。

【 0 0 6 3 】

CMIP において、インスタンスに対するフィルタ処理が必要な場合があり、このような場合に本実施例が適用できる。

【 0 0 6 4 】

この実施の形態によるスコープ処理方法によれば、スコープ条件に合致するインスタンスを全て抽出してから、フィルタ処理を行うのではなく、スコープ条件とフィルタ条件の双方に合致するインスタンスのみを抽出する。したがって、OSI 管理システム内に無駄なバッファ領域を持つ必要がなくなる。

【 0 0 6 5 】

【発明の効果】

本発明によるスコープ処理方法では、スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動しながら目的とするインスタンスを

抽出するので、包含木内の全範囲を移動する必要がない。したがって、短時間でスコープ処理を終了することができる。

【 0 0 6 6 】

また、本発明による O S I 管理システムにおける管理対象装置は、スコープ処理の開始位置とスコープ条件とによって定められる包含木内の範囲を移動しながら目的とするインスタンスを抽出するインスタンス抽出手段を備えた構成であるので、M I B の包含木の全範囲を移動しなくてもよい。したがって、短時間で管理対象情報を抽出することができる。

【図面の簡単な説明】

【図 1】 スコープ記述子オブジェクトを示す説明図である。

【図 2】 スコープ条件情報部を示す説明図である。

【図 3】 包含木の例を示す説明図である。

【図 4】 スタック領域の変化の例を示す説明図である。

【図 5】 スコープ記述子オブジェクトの生成動作の例を示すフローチャートである。

【図 6】 スコープ記述子オブジェクトが条件に該当するインスタンスを抽出する動作の例を示すフローチャートである。

【図 7】 深さ優先探索方法による移動の例を示したフローチャートである。

【図 8】 上方向および下方向への移動可否判断の例を示すフローチャートである。

【図 9】 特定のインスタンスを抽出するスコープ記述子オブジェクトを示す説明図である。

【図 1 0】 スコープ記述子オブジェクトがフィルタ条件等に該当するインスタンスを抽出する動作の例を示すフローチャートである。

【図 1 1】 O S I 管理システムの構成の例を示すブロック図である。

【符号の説明】

- 2 スコープ記述子オブジェクト
- 3 スコープ条件情報部
- 4 スコープ処理振舞部

5 スタック領域

7 M I B

1 0 0 管理対象装置

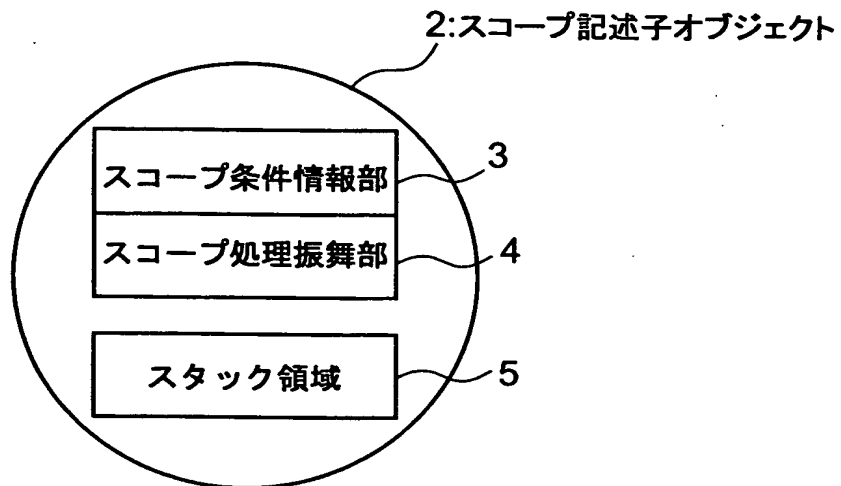
1 0 1 制御部

1 0 2 記憶装置

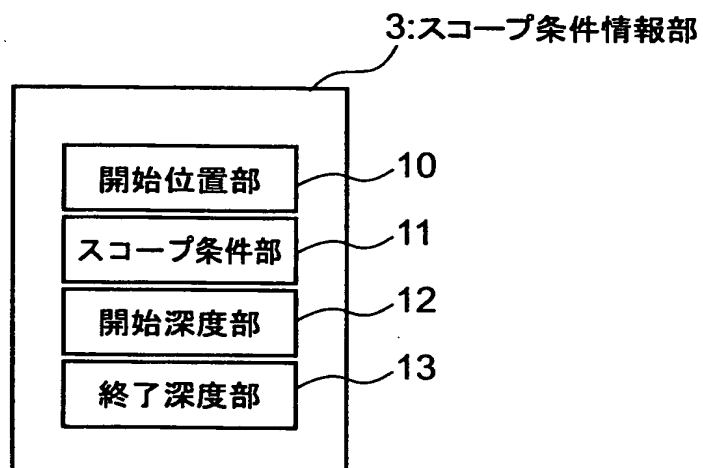
1 0 3 エージェント

【書類名】 図面

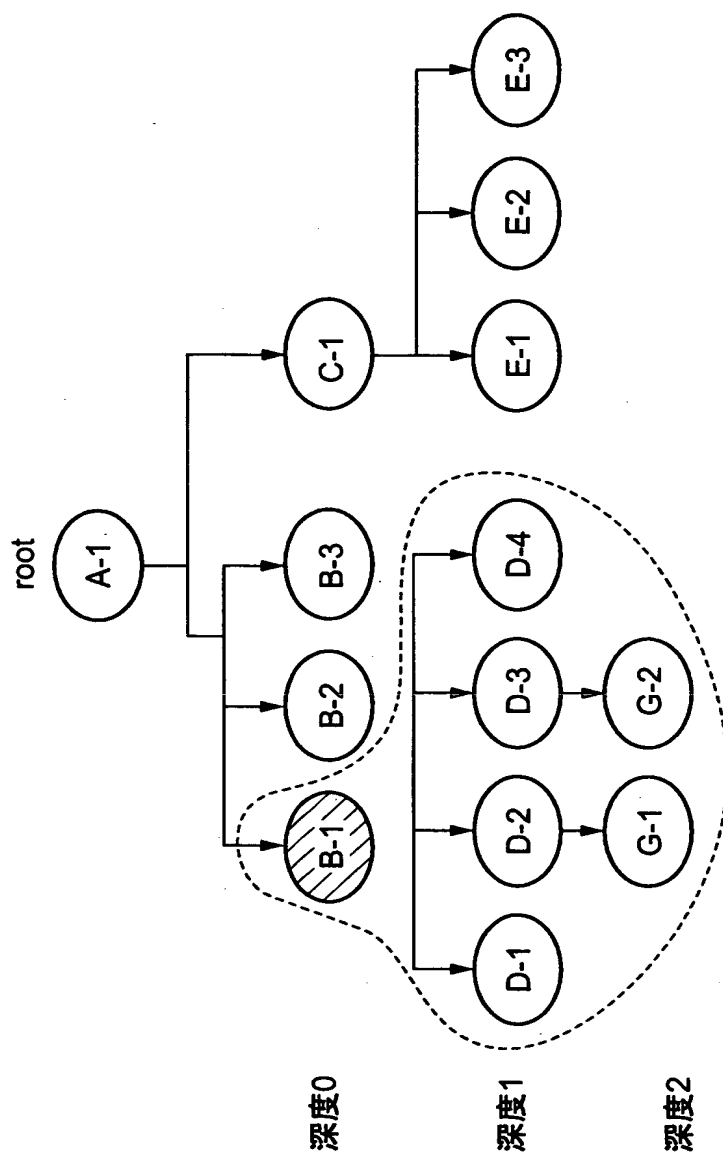
【図 1】



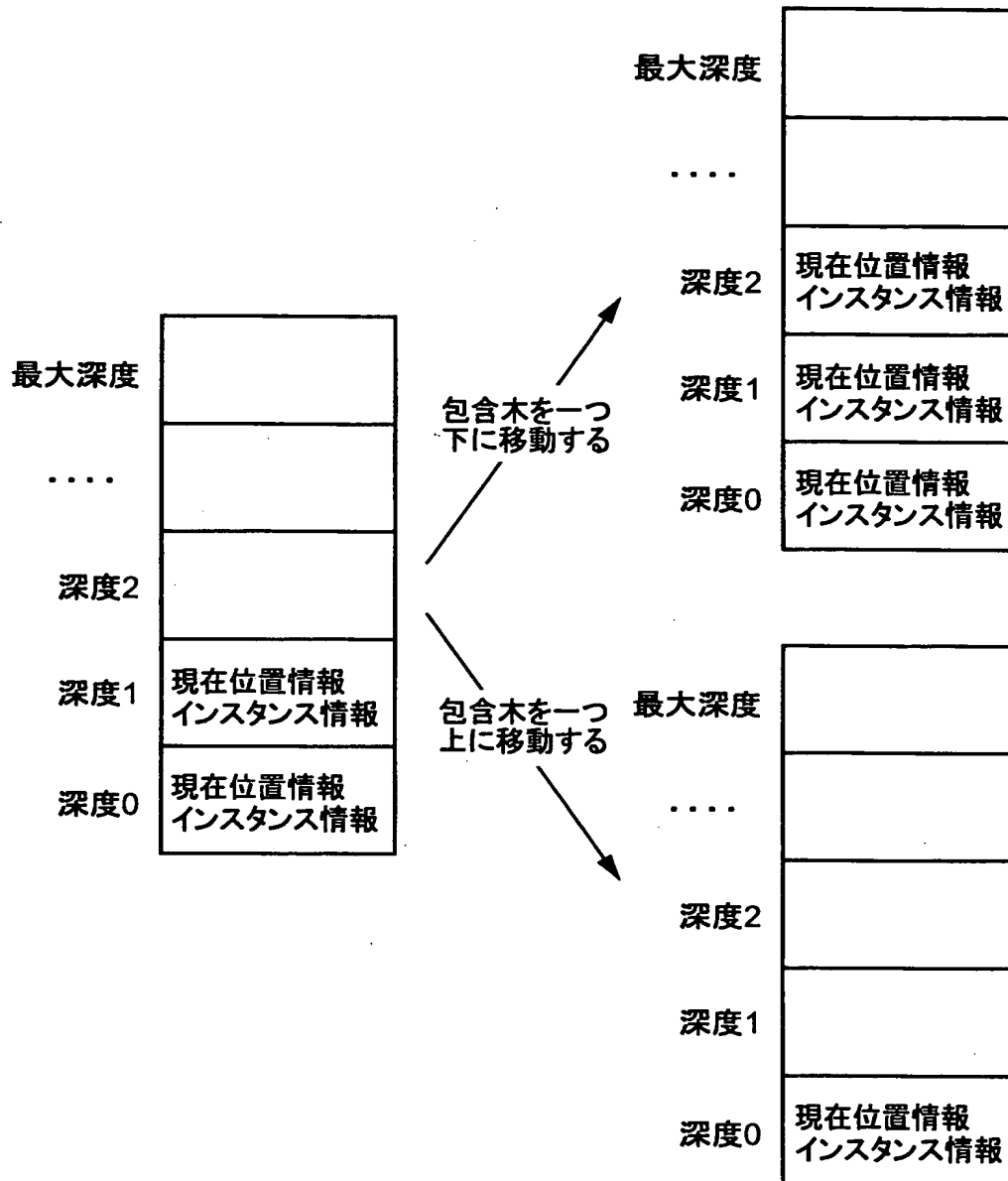
【図 2】



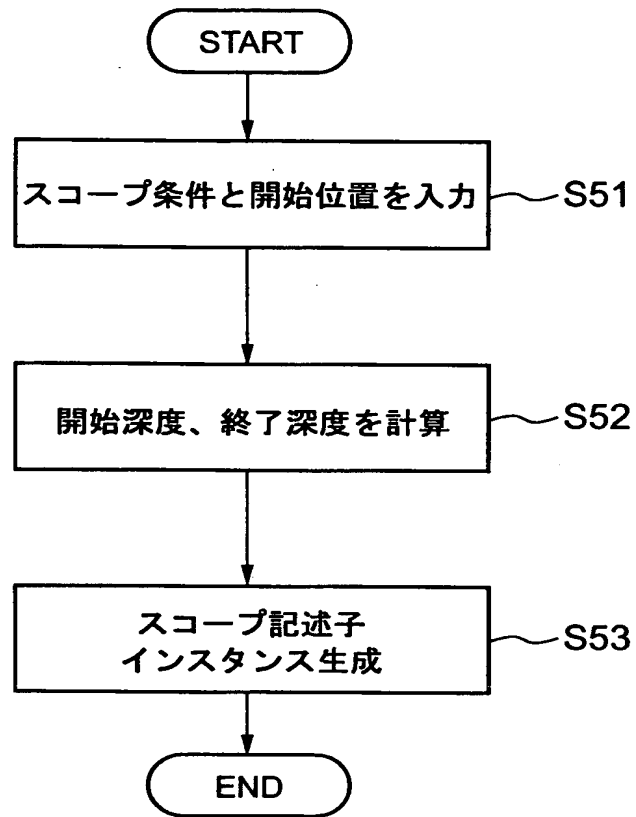
【図 3】



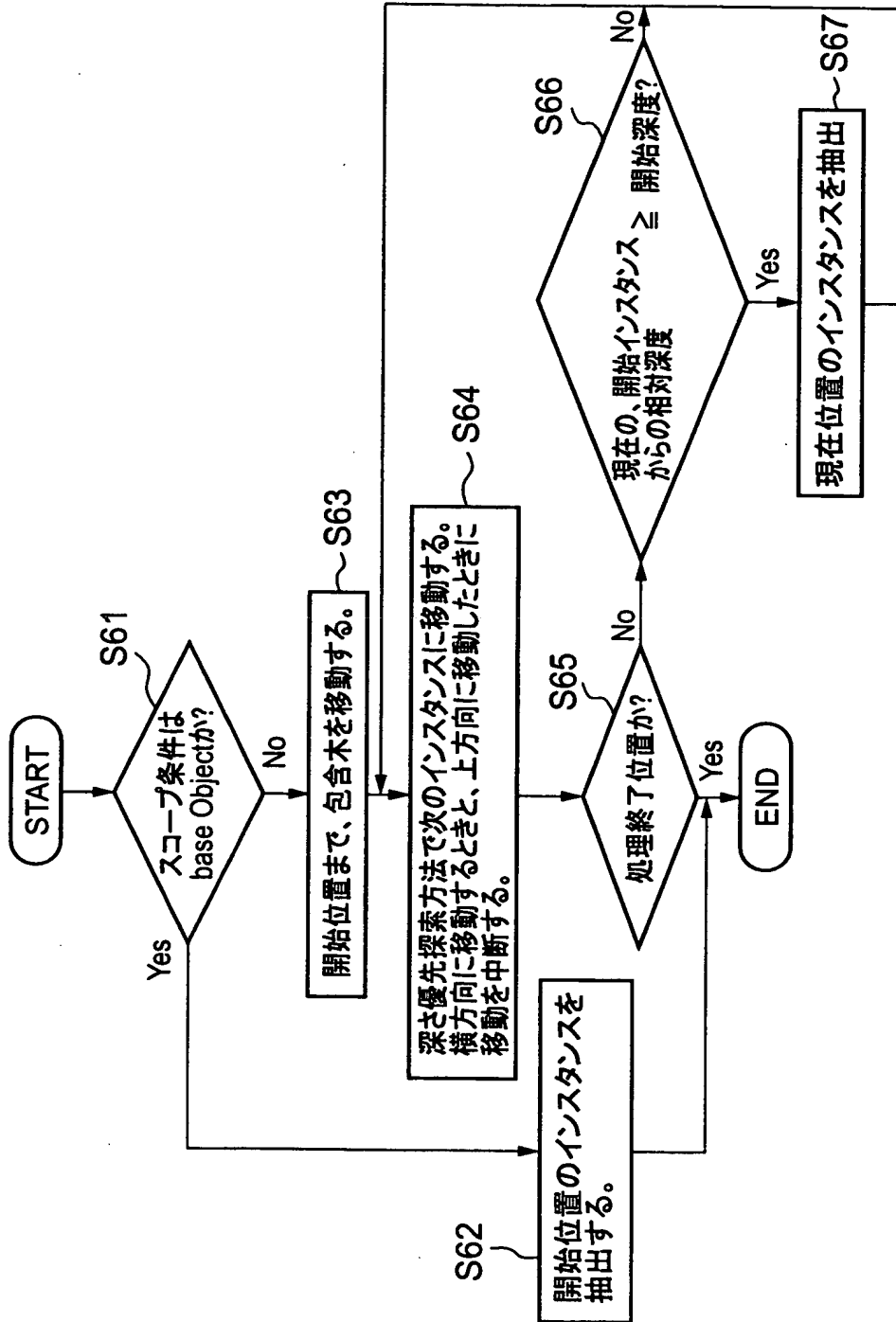
【図 4】



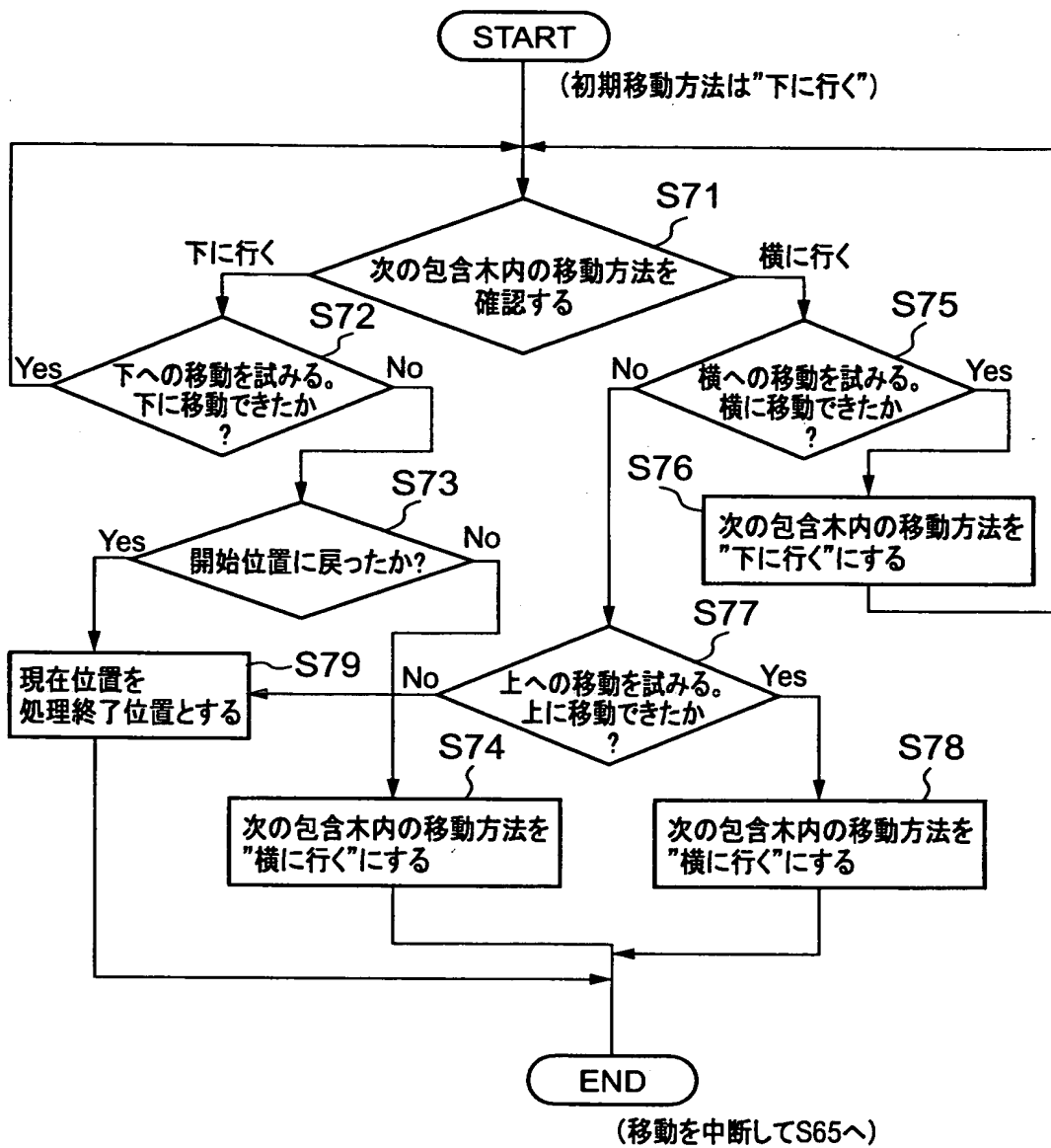
【図 5】



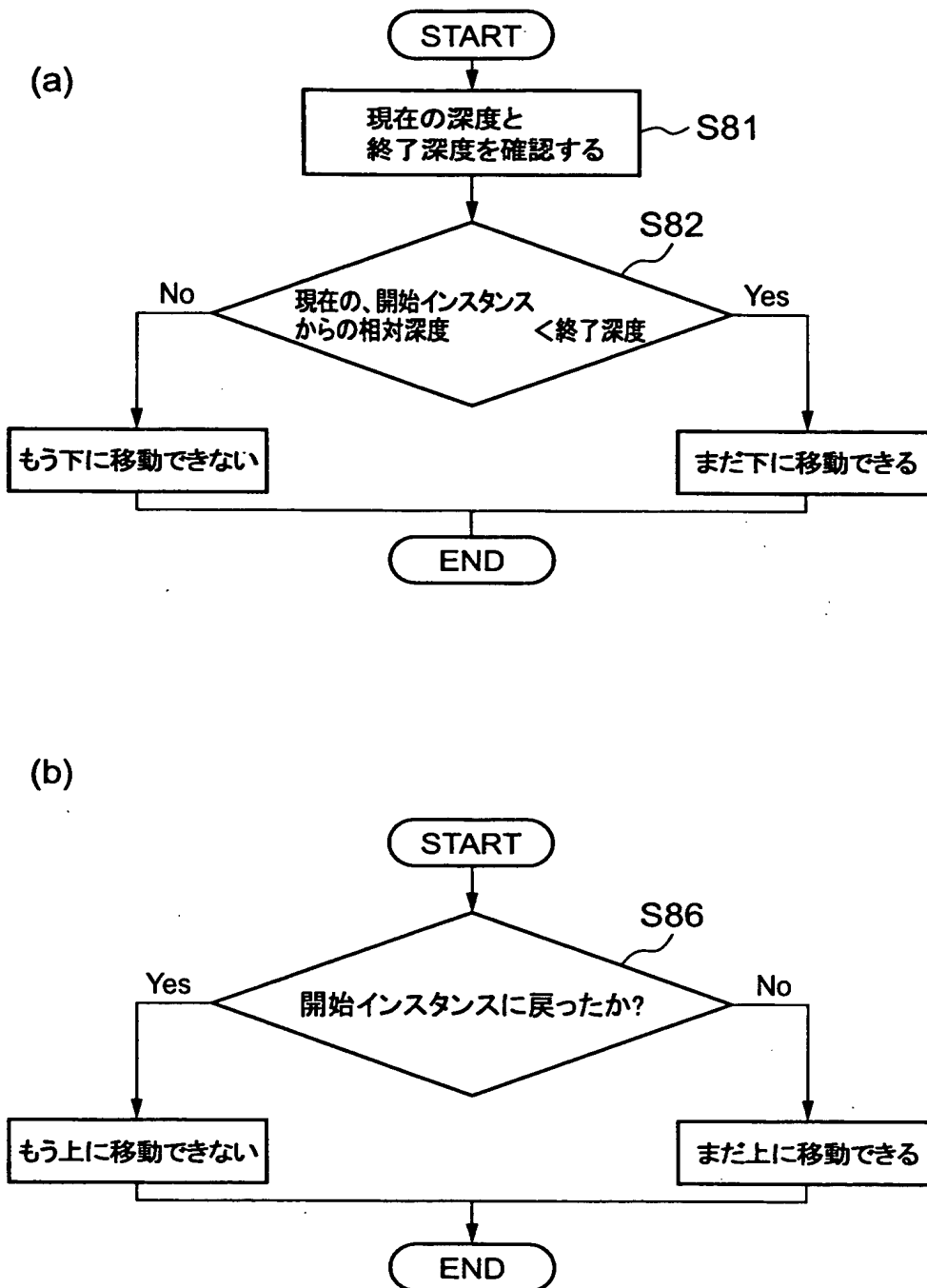
【図 6】



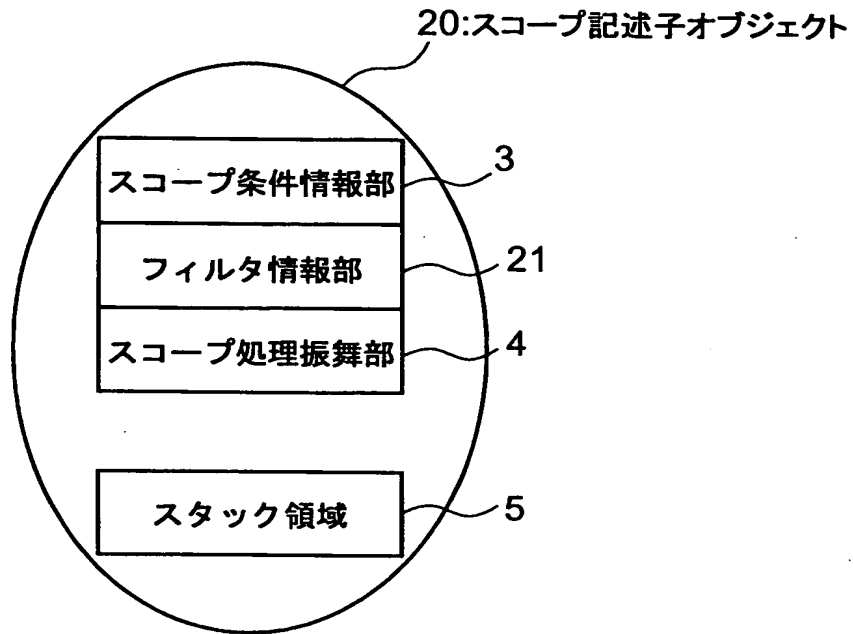
【図 7】



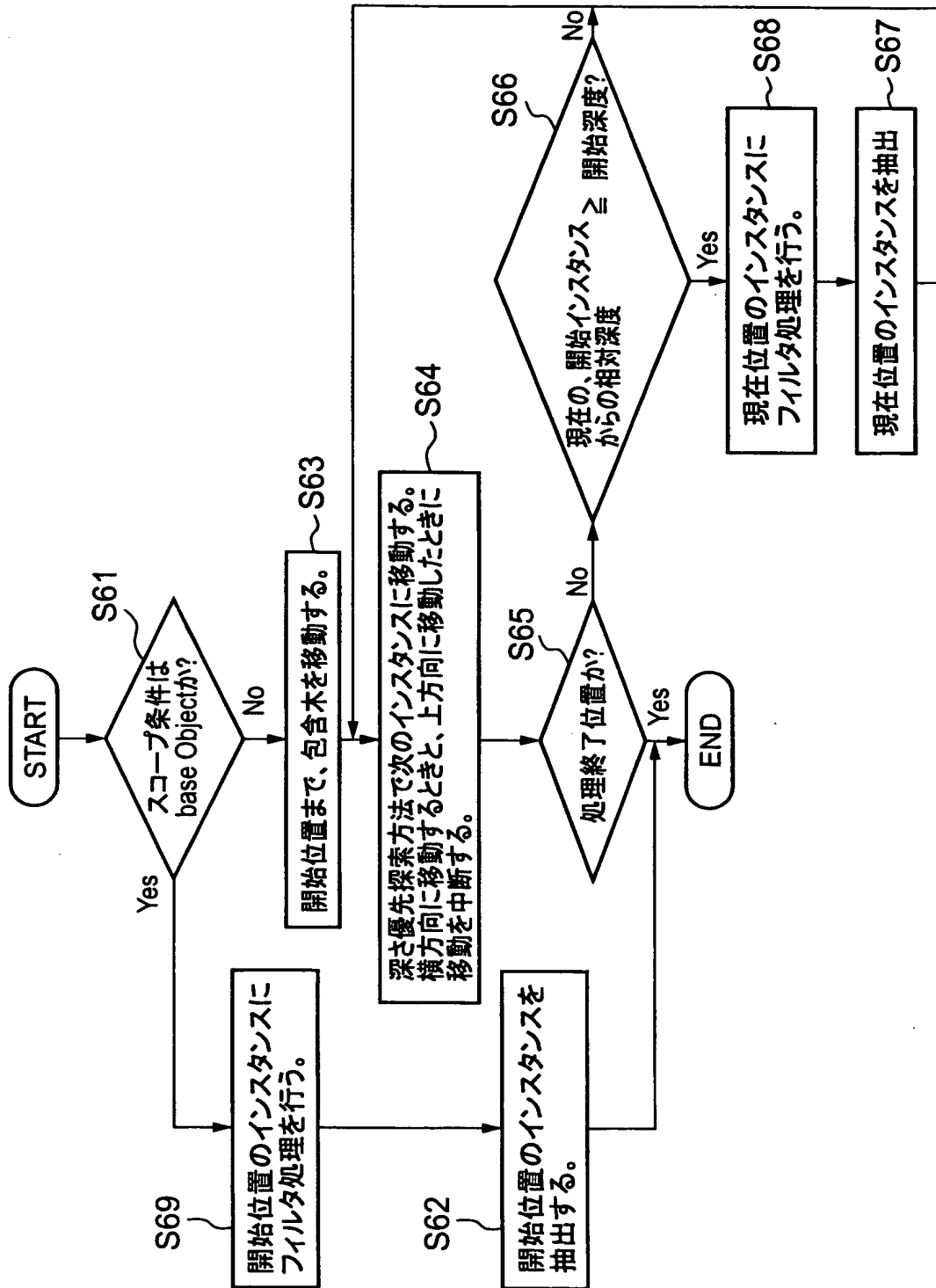
【図 8】



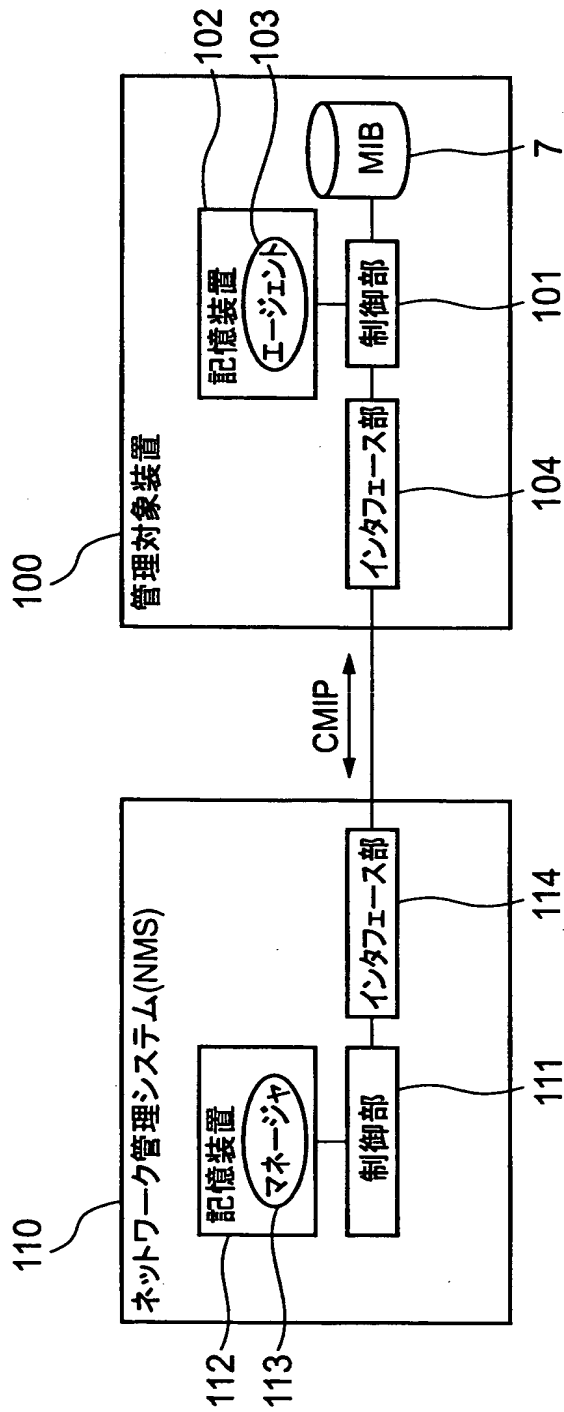
【図 9】



【図10】



【図 11】



【書類名】 要約書

【要約】

【課題】 包含木から短時間でインスタンスを抽出する。

【解決手段】 包含木に対してスコープ処理を行う場合、スコープ処理開始位置とスコープ条件によって定められる範囲を移動しながら目的とするインスタンスを抽出する。包含木内を移動するときには、まず、下位に移動する（ステップ S 7 2）。下位に移動できなくなったら移動方向を横に変え（ステップ S 7 4）、横に移動できたならば（ステップ S 7 5）、移動方向を下に変え（ステップ S 7 6）、下位へ移動していく。横に移動できなくなったら、上位に移動して（ステップ S 7 7）、横への移動を試みる。下位に移動できなくなったときと、横に移動できずに上位に移動したときに、移動を中断する。移動を中断したインスタンスが目的とするインスタンスであるならば、そのインスタンスを抽出する。

【選択図】 図 7

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社